

dp_spi_psd

User Manual

Version 1.3.1
21 August 2002

Jürgen Knödseder
Centre d'Etude Spatiale des Rayonnements
knodlseder@cesr.fr
<http://www.cesr.fr/~jurgen/index.html>

Note to the user

This software has been written to analyse data of the SPI telescope onboard INTEGRAL. Particular care has been taken in making the software user friendly and well documented. If you appreciated this effort, and if this software and User Manual were useful for your scientific work, the author would appreciate a corresponding acknowledgment in your published work.

Contents

1	Introduction	1
2	Getting started	1
3	Parameter file	3
4	Interface definition	4
5	Error codes	7

1 Introduction

The executable `dp_spi_psd` is part of the Data Preparation module of the ISDC system. It decodes the scientific event word of PSD events (PE) and PSD curves (CRVE), characterises all received PSD curves using the standard pulse shape characterisation procedure, and performs pulse shape fitting for quality control for a single science window.

For each PE, `dp_spi_psd` fills columns in the `SPI.-xPSD-PRP` data structures, where `x` is one of `O` or `D` and indicates the SPI scientific mode (Operational, Diagnostics). For each CRVE, `dp_spi_psd` fills columns in the `SPI.-xCRV-PRP` data structure, where `x` is one of `O`, `C`, or `D` and indicates the SPI scientific mode (Operational, Calibration, Diagnostics). The executable loops over all science modes that create PE and CRVE and treats all events and curves that are found.

It is assumed that the PRP data structures exist already and that the OBT column is filled. The number of rows in the `RAW` and `PRP` data structures must be identical, otherwise a warning will be issued and the corresponding science mode will not be handled. Empty or absent data structures are not treated.

Pre-existing entries in the PRP data structures are overwritten by the task. Hence `dp_spi_psd` may be applied repeatedly to the same data if for example some input parameters were erroneous in a previous task.

For the decoding of the PSD scientific event word, the PSD configuration has to be known. `dp_spi_psd` determines the PSD configuration throughout the entire science window using the telecommand report telemetry that is periodically downlinked each 3840 seconds or on request. Since this periodicity exceeds the typical duration of a science window, `dp_spi_psd` search also preceeding science windows for the PSD configuration in order to assure a complete coverage. To make this functionality working, the ISDC environment variable `REP_BASE_PROD` has to be set to the correct data repository. If the repository is not set the program will alert the user but continue execution.

`dp_spi_psd` is an ANSI C++ executable that has been developped under ISDC support platform 4.1 and requires the libraries `spi_psdlib` version 1.5.0 and `spi_toolslib` version 1.8.0 or higher.

2 Getting started

Before installing `dp_spi_psd`, make sure that the ISDC support platform 4.1 or higher is installed on your system, and that the libraries `spi_psdlib` 1.5.0 and `spi_toolslib` 1.8.0 or higher are installed.

After downloading the `dp_spi_psd.tar.gz` file, step into a directory that should hold the distribution, move the `dp_spi_psd.tar.gz` file into this directory and type:

```
$ gunzip dp_spi_psd.tar.gz
$ tar xvf dp_spi_psd.tar
```

The first command uncompresses the distribution file, the second unpacks the files.

Before configuration, the distribution needs to be reset to a clean state. To do this, type

```
$ make distclean
```

Then, configure the distribution. It is assumed here that you have previously installed the ISDC support platform, thus you should type

```
$ ~/bin/ac_stuff/configure
```

Finally, build the distribution by typing

```
$ make global_install
```

To perform a unit test, type

```
$ make test
```

Make sure that the test data `spi_test_data-1.0.tar.gz` are available at your site (they should reside in a directory whose name is defined by the `ISDC_TEST_DATA_DIR` environment variable).

3 Parameter file

```
#####
#
#           Centre d'Etude Spatiale des Rayonnements
#           (in collaboration with ISDC)
#
#           DATA PREPARATION
#
# -----
#
# File:      dp_spi_psd.par
# Version:   1.3.1
# Component: dp
#
# Author:    Juergen Knoedlseder
#            knodlseder@cesr.fr
#            CESR
#
# Purpose:   Parameter file of the SPI PSD data preparation
#            executable
#
# History:   1.3.1 21-Aug-2002 First ISDC delivery (Rev. 3)
#
#####
#
# Input SWG to process
#=====
swgdol, s, ql, "swg_prp.fits[1]",,,"ScW Group DOL (level PRP)"
#
# Library and algorithm parameter DOL
#=====
algopardol, s, hl, "psdFM280701.fits[SPI.-ALGO-PSD]",,,"PSD Algorithm parameters DOL"
libdol,      s, hl, "psdFM280701.fits[SPI.-LIB.-PSD]",,,"PSD on-board library filename"
#
# Standard parameters
#=====
clobber, b, hl, no,,, "Clobber Flag"
mode,    s, hl, "ql",,,"Execution mode"
```

The following parameters have to be specified:

- **swgdol**: specifies a science window group of level PRP that holds all data structures that are read or written by the task. Note that the task will search for PSD configurations in previous science windows. Only the RAW level of these previous science windows has to exist.
- **algopardol**: the DOL of the actually uplinked PSD algorithm parameters.
- **libdol**: the DOL of the actually uplinked pulse shape template library. If the library does not correspond to the uplinked library, alerts will eventually be generated by the executable.
- **clobber**: standard parameter (see Common User Manual)
- **mode**: standard parameter (see Common User Manual)

4 Interface definition

dp_spi_psd works on individual science windows, and as input DOL a PRP level science window group has to be provided. No element is attached by this task to this group, **it is assumed that all HDUs exist prior to task execution.**

To prepare the PSD data, dp_spi_psd needs the actual configuration of the PSD onboard library, which is usually delivered to ISDC in parallel as new library uplink commands are sent to MOC. The actual PSD onboard libraries may also be downloaded from the web site <http://www.cesr.fr/~jurgen/psd>. Together with the PSD libraries, PSD algorithm parameters are uplinked in form of a special library template to the PSD sub-assembly. Also these parameters are required for proper onground pulse shape analysis, and also these data are available at ISDC or at the web site. Since these data should change rather rarely (about each 6 months), no automatic indexing scheme has been implemented, but **the DOLs have to be changed manually by the operator after a PSD library uplink.**

The PSD configuration is also needed for PSD data preparation, but this configuration is reported cyclically each 3840 seconds in the telemetry (or even more often on request). Since this period exceeds the typical duration of a science window, preceeding science windows are also searched for PSD configuration information in order to gather the complete information about the PSD configuration throughout the entire science window. **Thus if the PSD configuration changes during the science window, dp_spi_psd will automatically take this into account for the decoding and the analysis of the PSD data and no manual interaction is needed.**

Preceeding science windows are localised by reading the "PREVSWID" attribute of the science window group, and by localising the DOL using the DAL3GEN function DAL3GENopenGroupID. For proper functioning the environment variable REP_BASE_PROD has to be set to the data repository that should be searched for preceeding science windows. If this environment variable is not set the task will alert the user but continue execution. In this case it may happen that the first events of the science window are not covered by a valid PSD configuration, hence no data preparation will be performed on them. This case will be signalled by setting the PSD_ERR column of the PRP data structures to the error code PSDLIB_ERRCODE_NO_CONFIGURATION.

The following list provides all possible error codes that may appear in the PSD_ERR column (see also the spi_psdlib user manual):

```
// 0 - 15 : onboard errors
#define PSDLIB_ERRCODE_NO_VALID_LIBRARY      0 // no valid library
#define PSDLIB_ERRCODE_PULSE_SATURATED      1 // saturated pulse
#define PSDLIB_ERRCODE_PULSE_AREA_TOO_SMALL 2 // pulse area too small
#define PSDLIB_ERRCODE_ATTP_TOO_EARLY      3 // ATTP too early
#define PSDLIB_ERRCODE_ATTP_TOO_LATE       4 // ATTP too late
#define PSDLIB_ERRCODE_BASELINE_TOO_LOW    5 // baseline too low
#define PSDLIB_ERRCODE_PULSE_STARTS_IN_BASELINE 6 // start in baseline
#define PSDLIB_ERRCODE_PULSE_ENDS_IN_BASELINE 7 // ends in baseline
#define PSDLIB_ERRCODE_PULSE_ENDS_TOO_LATE 8 // ends too late
#define PSDLIB_ERRCODE_PULSE_DURATION_TOO_SHORT 9 // duration too short
#define PSDLIB_ERRCODE_PULSE_DURATION_TOO_LONG 10 // duration too long
#define PSDLIB_ERRCODE_INVALID_DETECTOR    11 // invalid detector ID
#define PSDLIB_ERRCODE_PULSE_AREA_ZERO     12 // pulse area zero
#define PSDLIB_ERRCODE_BASELINE_TOO_HIGH   13 // baseline too high
#define PSDLIB_ERRCODE_BASELINE_OUTLIER    14 // baseline outlier
#define PSDLIB_ERRCODE_PULSE_AREA_TOO_LARGE 15 // pulse area too large
// 128 : no error
#define PSDLIB_ERRCODE_NO_ERROR             128 // no error
// 130 - 133 : onground science word decoding
#define PSDLIB_ERRCODE_NO_CORRELATION       130 // correlation failure
```

```
#define PSDLIB_ERRCODE_BAD_DECODING          131 // bad decoding
```

Error codes from 0-15 represent onboard error codes that were set by the PSD sub-assembly. They have the following meaning:

- **PSDLIB_ERRCODE_NO_VALID_LIBRARY**: signals that no valid library was available for pulse shape analysis. This error may occur in case of a PSD configuration problem/error.
- **PSDLIB_ERRCODE_PULSE_SATURATED**: the pulse shape was saturated and hence not analysed (this should regularly occur for PSD events with energies above about 2 MeV)
- **PSDLIB_ERRCODE_PULSE_AREA_TOO_SMALL**: the pulse shape area - which is a coarse measure of the event energy - was smaller than a defined limit (the limit is defined by the PSD algorithm parameters and may serve as lower energy threshold)
- **PSDLIB_ERRCODE_ATTP_TOO_EARLY**: the pulse peaks too early for a proper pulse shape analysis. This may happen in case of pulse pile up or if the PSD trigger threshold is set too high.
- **PSDLIB_ERRCODE_ATTP_TOO_LATE**: the pulse peaks too late for a proper pulse shape analysis. This may happen in case of noise triggers or pile up.
- **PSDLIB_ERRCODE_BASELINE_TOO_LOW**: the pulse baseline is below a threshold that is defined by the PSD algorithm parameters. This threshold may help to eliminate invalid PSD triggers.
- **PSDLIB_ERRCODE_PULSE_STARTS_IN_BASELINE**: the pulse starts in a domain that has been used for baseline determination, hence the baseline may have not been determined correctly and the pulse shape analysis may be deteriorated. This should only happen in case of noise triggers.
- **PSDLIB_ERRCODE_PULSE_ENDS_IN_BASELINE**: the pulse ends in a domain that has been used for baseline determination, hence the baseline may have not been determined correctly and the pulse shape analysis may be deteriorated. This should only happen in case of noise triggers.
- **PSDLIB_ERRCODE_PULSE_ENDS_TOO_LATE**: the pulse ends after the PSD pulse acquisition was stopped, hence the end of the pulse is truncated and the pulse shape analysis may be deteriorated. This should only happen in case of noise triggers or eventually pile-up.
- **PSDLIB_ERRCODE_PULSE_DURATION_TOO_SHORT**: the pulse duration is shorter than a limit that is defined by the PSD algorithm parameters. This limit may help to eliminate invalid PSD triggers.
- **PSDLIB_ERRCODE_PULSE_DURATION_TOO_LONG**: the pulse duration is longer than a limit that is defined by the PSD algorithm parameters. This limit may help to eliminate invalid PSD triggers.
- **PSDLIB_ERRCODE_INVALID_DETECTOR**: the PSD has detected a detector ID that is not comprised between 0 - 18. This may indicate a hardware problem.
- **PSDLIB_ERRCODE_PULSE_AREA_ZERO**: the pulse has a vanishing area. This may happen in case of noise triggers.
- **PSDLIB_ERRCODE_BASELINE_TOO_HIGH**: the pulse baseline is above a threshold that is defined by the PSD algorithm parameters. This threshold may help to eliminate invalid PSD triggers.
- **PSDLIB_ERRCODE_BASELINE_OUTLIER**: the pulse has a baseline that differs substantially from the baselines of the preceeding pulses, hence it is included. This may happen after a ULD event.
- **PSDLIB_ERRCODE_PULSE_AREA_TOO_LARGE**: the pulse shape area - which is a coarse measure of the event energy - was larger than a defined limit (the limit is defined by the PSD algorithm parameters and may serve as upper energy threshold)

Error code 128 corresponds to no error. **This should be the standard error code and signals a healthy status of the PSD science data.**

Error codes 130-133 may occur during science word decoding and they have the following meaning:

- **PSDLIB_ERRCODE_NO_CORRELATION**: is an onboard error that signals that the PSD correlation was disabled in the SPI DPE. This can happen in some special configurations but should of course not be the rule since in this case no PSD information is available onground.
- **PSDLIB_ERRCODE_BAD_DECODING**: signals that the detected PSD configuration was incompatible with the encoding of the PSD science data, hence decoding was not possible (this error may occur if the number of templates used for the PSD library analysis is either 0 or larger than 180). The error may occur if some erroneous PSD configuration was read.

Pulse fitting may also lead to some error codes that are stored in the **PSD_FIT_ERR** column of the **DRP** data structure. The following codes are defined (see also the **spi-psdlib** user manual):

```
// 128 : no error
#define PSDLIB_ERRCODE_NO_ERROR          128 // no error
// 140 - 143 : onground fitting
#define PSDLIB_ERRCODE_FIT_NO_CURVE      140 // no curve available
#define PSDLIB_ERRCODE_FIT_TOO_FEW_TIME_STEPS 141 // too few time steps
#define PSDLIB_ERRCODE_FIT_TOO_MANY_TIME_STEPS 142 // too many time steps
#define PSDLIB_ERRCODE_FIT_TOO_MANY_TEMPLATES 143 // too many templates
#define PSDLIB_ERRCODE_FIT_STARTTIME_TOO_LATE 144 // starttime too late
#define PSDLIB_ERRCODE_FIT_NEGATIVE_BASELINE 145 // negative baseline
#define PSDLIB_ERRCODE_FIT_NEGATIVE_PULSE_AREA 146 // negative pulse area
```

Error code 128 corresponds to no error. **This should be the standard error code and signals a healthy status of the PSD pulse fitting.**

Error codes 140-146 signal a problem during onground curve fitting and they have the following meaning:

- **PSDLIB_ERRCODE_FIT_NO_CURVE**: no PSD curve was available (may indicate a memory allocation problem)
- **PSDLIB_ERRCODE_FIT_TOO_FEW_TIME_STEPS**: the selected number of time steps was inferior to 6. There is a problem with the detected PSD configuration.
- **PSDLIB_ERRCODE_FIT_TOO_MANY_TIME_STEPS**: the selected number of time steps was superior to the maximum allowed number that is defined by the PSD algorithm parameters. Either there is a problem with the detected PSD configuration or the algorithm parameter DOL is incorrect (i.e. it does not correspond to the uplinked parameters)
- **PSDLIB_ERRCODE_FIT_TOO_MANY_TEMPLATES**: the selected number of templates exceeds the available number. Either there is a problem with the detected PSD configuration or the algorithm parameter DOL is incorrect (i.e. it does not correspond to the uplinked parameters)
- **PSDLIB_ERRCODE_FIT_STARTTIME_TOO_LATE**: the pulse starts too late for a proper analysis. The pulse was possibly a triggered by noise.
- **PSDLIB_ERRCODE_FIT_NEGATIVE_BASELINE**: the pulse had a negative baseline, hence fitting was impossible
- **PSDLIB_ERRCODE_FIT_NEGATIVE_PULSE_AREA**: the pulse had a pulse area, hence fitting was impossible

The task `dp_spi_psd` handles all data in junks of moderate size in order to limit the memory requirements. The junk size for PSD events and PSD curves are set by the two definitions

```
#define DP_SPI_PSD_BUFFER_PE    100000  // PE junk size
#define DP_SPI_PSD_BUFFER_CRV  100000  // Curve junk size
```

in the task header file. The user/operator may change these junk sizes if it becomes necessary.

5 Error codes

The following error codes that are proper to `dp_spi_psd` may occur during task execution.

```
DP_SPI_PSD_ERROR_MEM_ALLOC  -231600 // Dynamical memory allocation failure
DP_SPI_PSD_ERROR_NO_CFG     -231601 // No PSD configuration error
```

They have the following meaning:

- `DP_SPI_PSD_ERROR_MEM_ALLOC` : the allocation of dynamical memory has failed. Probable your system resources are too limited to run this task. If you cannot increase your resources you may reduced the numbers `DP_SPI_PSD_BUFFER_PE` and `DP_SPI_PSD_BUFFER_CRV` in the `dp_spi_psd.h` header file and recompile the code.
- `DP_SPI_PSD_ERROR_NO_CFG` : not used anymore.

In addition, all errors that may occur in calls to ISDC support functions (such as for example DAL, RIL or PIL) are forwarded. Please consult the ISDC web pages for getting information about these error codes.